2178

Attorney's Docket No.: 06975-124001 / Multimedia 15

SC
#43
4-12 04

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant	: David Corboy	Art Unit	: 2178
Serial No.	: 08/866,857	Examiner	: Cong-Lac Huynh
Filed	: May 30, 1997		
Title	: ENCAPSULATED DOCUMENT AND FORMAT SYSTEM		

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF ORIGINAL DECLARATION UNDER 37 C.F.R. §1.131

Further to the Amendment filed on March 22, 2004, Applicant submits herewith the **original** executed Declaration Under 37 C.F.R. §1.131 for entry into the above-identified application.

Please apply any charges or credits to Deposit Account No. 06-1050.

Respectfully submitted,

Date: March 31, 2004

[illegible]
Scott R. Boalick
Reg. No. 42,337

Fish & Richardson P.C.
1425 K Street, N.W.
11th Floor
Washington, DC 20005-3500
Telephone: (202) 783-5070
Facsimile: (202) 783-2331

40211909.doc

Attorney's Docket No.: 06975-124001 / Multimedia 15

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant : David Corboy		Art Unit	: 2178
Serial No. : 08/866,857		Examiner	: Cong-Lac Huynh
Filed : May 30, 1997			
Title : ENCAPSULATED DOCUMENT AND FORMAT SYSTEM			

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

DECLARATION UNDER 37 C.F.R. §1.131

I, David Corboy, hereby declare as follows:

1. I have read and understood the text of U.S. Application No. 08/866,857 (the '857 application), which discloses an invention for which I am the inventor.

2. On or prior to April 6, 1997, I implemented and practiced the methods and computer systems described in paragraph 6 of this document.

3. The attached pages are photocopies of:

a) A redacted draft patent application dated at least as early as April 6, 1997 (Exhibit 1) describes an encapsulated document and format system. The '857 application includes figures and description that includes features described by the draft application.

The system described in Exhibit 1 corresponds to the methods and computer systems claimed in the '857 application.

4. The draft patent application listed in paragraph 6 was produced or written by me and/or produced or written at my direction by my patent attorney on or prior to April 6, 1997. The draft patent application was produced in the ordinary course of business in the United States.

5. The system described in the draft patent application of Exhibit 1 includes the method and computer system described in paragraph 6 of this document.

6. With respect to independent claims 1 and 10 of the '857 application, I implemented and practiced a method and a computer system that received a stream including a file that integrates media content with choreography information within each of at least two objects of the file (e.g., Exhibit 1 at page 3, line 22 to page 4, line 2; page 4, lines 8-13; page 4, line 23 to page 5, line 7; page 8, lines 5-20). Each of the objects included media content data and choreography information associated therewith (e.g., Exhibit 1 at page 18, lines 8-19). The choreography information included data indicating an author-designated relationship between the objects of the file that defines an author-designated temporal order of presentation between the objects (e.g., Exhibit 1 at page 18, lines 13-19). Before all objects of the file were received, the method and computer system began to render media content encapsulated within the file based on the choreography information associated with objects received (e.g., Exhibit 1 at page 3, lines 11-17; page 17, line 23 to page 18, line 7; page 22, lines 11-19). Display of the objects received was enabled based on the temporal order defined by the choreography information (e.g., Exhibit 1 at page 18, lines 13-19). The temporal order was maintained independent of a recipient or a web server and independent of a bandwidth of a communications channel used to send the multimedia document (e.g., Exhibit 1 at page 17, line 23 to page 19, line 19).

Also, with respect to independent claim 100, I implemented and practiced a method that received specification of media content by an author and received designation by the author of choreography information that indicates at least an intended order of presentation for the specified media content was received (e.g., Exhibit 1 at page 22, lines 11-19). A single file that integrated the media content with the choreography information was generated (e.g., Exhibit 1 at page 3, line 22 to page 4, line 2; page 4, lines 8-13; page 4, line 23 to page 5, line 7; page 8, lines 5-20). Generating the single file included encapsulating within the single file at least two objects, where each object included media content data and choreography information associated therewith (e.g., Exhibit 1 at page 18, lines 8-19). The choreography information included data defining an author-designated relationship between the objects of the single file that defined an author-designated temporal order of presentation between the objects (e.g., Exhibit 1 at page 18, lines 13-19). Before all objects of the file were received by a recipient, the method enabled the recipient to begin rendering the media content encapsulated within the file according to the temporal order defined by the choreography information associated with objects received (e.g., Exhibit 1 at page 3, lines 11-17; page 17, line 23 to page 18, line 7; page 22, lines 11-19). The temporal order was maintained independent of a recipient or a web server (e.g., Exhibit 1 at page 17, line 23 to page 19, line 19).

The method, computer program and apparatus also implemented the subject matter of dependent claims 2-9, 11, 13-16, 31-50, 63-66, and 101-109 of the '857 application.

I further declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. §1001 and that such willful false statements may jeopardize the validity of the application or any patents issued thereon.

Signed and Declared at AOL'S STERLING, VA CAMPUS ON BRODERICK DRIVE this 19TH day of March, 2004

[illegible]
David Corboy

40209562.doc

FISH & RICHARDSON P.C.

4225 Executive Square
Suite 1400
La Jolla, California
92037

Telephone
619 678-5070

Facsimile
619 678-5099

Frederick P. Fish
1855-1930

W.K. Richardson
1859-1951

BOSTON
HOUSTON
NEW YORK
SOUTHERN CALIFORNIA
SILICON VALLEY
TWIN CITIES
WASHINGTON, DC

David Corboy
America Online, Inc.

Re: Patent Application for:
"ENCAPSULATED DOCUMENT AND FORMAT SYSTEM"

Dear Dave:

Attached is a latest version of the patent application for the above-entitled invention.

APPLICATION FOR UNITED STATES PATENT IN THE NAME OF

David Corboy

of

America Online, Inc.

FOR

ENCAPSULATED DOCUMENT AND FORMAT SYSTEM

ENCAPSULATED DOCUMENT AND FORMAT SYSTEM

BACKGROUND

SUMMARY

Modem computer files may include a number of parts representing various kinds of information. The parts are specially coded in a way to represent the particular information. Moreover, each set of bits in modem computer files is coded in a unique way to represent a unique kind of information. For example, the ASCII coding system assigns a bit sequence to each alphanumeric character of the ASCII set. That bit sequence represents the character, however, only if one knows that it is ASCII.

Bits can also be used to represent image and sound data. For example, image data may be represented in BITMAP format, or in any number of compressed image data formats. The set of bits itself, however, has no meaning unless one knows the format. The ASCII code for a particular alphanumeric character could equally well represent a sequence of sound, or a portion of an image. Different kinds of files are necessary, therefore, to transfer different kinds of information.

A multimedia document includes many different kinds of information. One example of a multimedia document is found as hypertext markup languages on the worldwide web. This information is a document in the sense that, once downloaded, a document-like image appears on the screen. The screen shows rich text (*i.e.*, text that is coded in terms of its size, color, and font) as well as images that are intended to be located at different locations within the document. A worldwide web download can also include sound as part of the downloaded document.

The various kinds of information in a multimedia document, however, are each formed from a separate file that includes instructions about where the various pieces of the document should go. For example, an HTML document includes rich text information, along with commands to insert various additional information. Such additional information, typically GIF or JPEG image files, is separately downloaded and inserted into the text.

The original format for downloading HTML documents displayed the formed document only when it was completely downloaded.

Moreover, the download of HTML documents cannot be sequenced or choreographed because of the nature of the documents. Information is downloaded as it becomes available. For example, if the document includes multiple images, they may come in any order. The information may come in any various number of forms.

It is difficult for a computer receiving an HTML document to save the document, because of the previously described inherent nature of HTML documents. The document itself is a text document that includes commands for additional images. The images are not part of the HTML document, but rather separate sub-documents. The perceived display of an HTML documents is thus assembled from separate downloaded pieces of information. HTML documents cannot be easily saved, however, because no connection exists between the various pieces of information and the various sub-documents. Moreover, HTTP documents cannot be easily reassembled or resent. While proposals do exist for "encapsulated" HTML that would allow an HTML document and all of its component parts to be concurrently saved for later viewing or forwarding, these proposals suffer from a variety of restrictions, and none would allow for data to be choreographed by the author.

providing a system and method for forming files as well as a file format, each of which can include integrated information. The information is stored in a special form that allows parts to be viewed while forming the file.

The file can be repeatedly stored and sent, because the encapsulated format facilitates keeping the file structure intact after the user has viewed the file.

Alternatively, only the formatting data may be contained in the document encapsulation (*i.e.*, the size, position, and other display characteristics of the document, irrespective of the actual document data). The actual document data could then be stored in and retrieved from an external source, such as an URL, an HTML, or a cache, or from another source, such as a CD. The user can still save and resend the encapsulated document,

as the document data can be retrieved from the external sources and encapsulated into the document.

A format of the invention multiplexes different forms of different data together. The multiplexed data is in a streaming format, also known as a progressive rendering formats, especially when dealing with image formats. The streaming format data, once downloaded, causes the display to change on an incremental basis. Further downloading occurs simultaneously with the displaying of the file.

: a system that permits a number of different file formats to be encapsulated in a way that enables choreographing between the file elements. By choreographing, the present specification refers to allowing the author of the file format to select the relative times at which various components of the file are displayed.

Special techniques are used with so-called temporal data. temporal data includes data that must be presented to the user during a specific time span, for example, sound files. Once a sound file begins playing, it should be played uninterrupted to avoid an unnatural feel to the sounds. The temporal operations of the present invention are used as part of the choreography.

recognizes temporal data objects and creates a multiplexed format in which temporal presentations are accurately conveyed to the user.

In a first embodiment, is a method for producing a hierarchical data file for a multimedia document. The data file has a plurality of different file formats encapsulated within the data file and is capable of being displayed on a computer display. The method includes creating a document; creating a first file support object including information in a first file format, the first file support object being capable of supporting a plurality of first lower objects; supporting the first file support object by the document; creating a second file support object including information in a second file format different from the first file format, the second file support object being capable of supporting a plurality of second lower objects; supporting the second file support object by the document; and displaying the computer display document.

In another embodiment, a hierarchical data file structure that encapsulates a plurality of different file formats to form a multimedia document. The multimedia document is capable of being displayed on a display of a computer system. The data file includes a document that includes information for controlling the display. The data file also includes a first support object including information in a first file format. The first support object is supported by the document and is capable of supporting a plurality of first lower objects. Each first lower object is a lower level object than the first support object in the hierarchical data file structure. The data file also includes a second support object including information in a second file format different from the first file format. The second support object is also supported by the control object and is also capable of supporting a plurality of second lower objects. Each second lower object is a lower level object than the second support object in the hierarchical data file structure.

In still another embodiment, a method of constructing a frame for a framed image to be included as part of a multimedia document. The multimedia document encapsulates data in a plurality of file formats and is capable of being displayed on a display of a computer. The computer includes a player. The method comprises: placing an image into the multimedia document; receiving information about the image by the player; decoding the image information; sending the decoded image information to the multimedia document; and enclosing the decoded image in a frame in the multimedia document.

In another embodiment, a method for multiplexing data in a multiplex message that includes data in a plurality of file formats. The file formats are selected from a group of file formats including a textual format, an image format, and a sound format. The multiplex message forms at least a portion of a multimedia document and includes a plurality of object files, each object file being represented by at least one data slice. The method includes providing an object number counter in the data file indicating the number of object files following the object number counter in the data file. The method further includes providing a plurality of object descriptions, each object description describing a corresponding one of the object files. The method also includes providing a choreography group including the data slices of the object files interleaved in a predetermined manner.

BRIEF DESCRIPTION OF THE DRAWINGS

FIGURE 1 is a block diagram showing system architecture

FIGURE 2 shows the overall hierarchy of the layered parts of a document including the hierarchy of a framed image.

FIGURE 3 shows an overall display of a multimedia document formed according to the hierarchy of FIGURE 2.

FIGURE 4 is a flow diagram showing the steps by which a code segment receives image information.

FIGURE 5 shows the preferred file format

FIGURE 6 shows a multiplexed scheme

FIGURE 7 shows an example of choreography of a document

FIGURE 8 shows the default method for organizing a document

Like reference numbers and designations in the various drawings indicate like elements.

DETAILED DESCRIPTION

Throughout this description, the preferred embodiment and examples shown should be considered as exemplars, rather than as limitations.

I. System Architecture

The preferred overall system architecture is shown in FIGURE 1 and is designated generally by reference numeral 100. A bit stream 102 is input to the system 100. The input bit stream 102 represents a new special data format that is referenced in this specification as the "Art Format." The Art Format is an encapsulated data format that includes various types of data.

The bit stream 102 is input from an overall information provider 104, for example, an Internet provider such as America OnLine®, to the input function of a player wrapper 106 as the bit stream 102 is received. The player wrapper includes a player dynamic link library (DLL) 108. The player DLL 108 separates and commands play of the various information in the bit stream 102. The playing is carried out using a technique known as "Art Doc Extensions," which is also preferably embodied as a DLL. It will be understood that, although the playing systems are described as being embodied as DLLs, the playing systems alternatively could be embodied as dedicated hardware components, *e.g.*, digital signal processors. Other examples of the player wrapper 106 include a Netscape or ActiveX plug-in. Player wrappers 106 can also allow editing and creation of multimedia documents.

The preferred embodiment uses a plug-in, rich text format text processor, a module of the player wrapper 106. It will be recognized that the rich text processor may be replaced by another module and is not integral to the present invention. The preferred plug-in text processor is Paige®, which is commercially available from Datapak, Inc. Paige® carries out many different operations and allows other modules to communicate with Paige®, such that those modules can tell Paige® what operation to perform in various ways. Paige® can respond to commands to use various fonts, put text in various places within a multimedia document, and reserve various other places within the document for other items, such as image data or video. It will be understood, of course, that alternate software packages and systems could be used.

The Paige® text processor may perform the displaying of the document. In such a case, Paige® communicates with a currently selected window on the computer display.

The preferred framework includes various overheads, such as clipboard, command stacks for undo, and other well-known overhead systems.

II. Construction and Features of an ARTDOC Document

The multimedia document itself is ordered as a series of layered parts. FIGURE 2 shows the overall hierarchy 200 of the layered parts of a document (referred to herein as an "ARTDOC document") in accordance with the present invention. The highest level portion of the ARTDOC document is the document object itself 202. The document object 202 controls information for the entire active display window, including commands for background colors, enclosing windows, or the like.

Each document portion has a capability of supporting "children," which are subspecies of the ARTDOC document. The document object 202 shown in FIGURE 2 supports two children—a main text object 204 and a frame object 206. The main text object 204 can include other children, including separator children 207, hypertext children 208, and in-line children 210. The hypertext children 208 can include links to other documents, which can be ARTDOC document images or any other document. The in-line children 210 can include in-line bit maps 228, sounds 230, or other multimedia objects, which can be displayed as part of the ARTDOC document. The separator children 207 may provide visual lines to separate text.

The frame 206, as shown, also includes a number of children. For example, the frame 206 includes an image 214 within the frame 206, hyperlinks 216 regarding the image 214, a sound file 219, and a slide show 217. The framed image 206 may also include caption text 218, which may also include text within a frame 222, hyperlinks 224, and in-line multimedia objects 226 like the main text object 204.

FIGURE 3 shows an overall display for an ARTDOC document 300 formed according to the hierarchy of FIGURE 2. A number of objects may be embedded within the document 300, along with text 310. The Paige® word processing program, as part of its function, controls wrapping of lines, *etc.*, around the images displayed in the document 300. Paige®, as now commercially available, allows its areas to be set-up and controlled. Paige® also allows certain areas to be defined that are not Paige® areas, for example, an image and/or its caption.

The system uses these Paige® capabilities to create an exclusionary area 302 within the document 300. A frame object 304, in this case including an image 305, is located within the exclusionary area 302, which effectively forms an invisible frame around the frame object 304. The enclosed objects can include framed images, slide shows, framed text, sounds, separators, and hyperlinks, which are all described in detail below.

A similar operation is carried out for the captioning operation, by which a caption 306 is placed adjacent the object 304, but within the exclusionary area 302. In this example, a hyperlink 308 is positioned over the graphical image 305. The hyperlink 308 may be an area of the image object 305 that one can select, for example, by a mouse click, to determine something else about the image object 305.

The hierarchal structure 200 of FIGURE 2 facilitates moving positions of objects within the ARTDOC document. Each object within the ARTDOC document is formed of a number of objects and sub-objects. The sub-objects are children of the objects. The hierarchical system 200 according to the present invention, as described above, allows a higher level object to be moved together with all of its lower level objects (*i.e.*, sub-objects). In the example of FIGURE 2, frame 206, image 214, hyperlink 216, and caption text 218 (with its children, text 222, hyperlinks 224, and in-line multimedia objects 226) are moved together with the frame 206 when it is moved. Therefore, for example, moving a frame will move not only the image within that frame, but also the hyperlinks and text within the frame. This same operation can be used with a delete command. Deleting the frame object deletes the frame, the image, the hypertext, and all of the other associated parts.

Each action (*e.g.*, a move command) is passed to the object, and the object decides what to do with that action. Thus, for example, to move a frame 206, a mouse operation is used to move the position of the frame 206 from a first position to a second position. The mouse operation is handled in the standard way by the operating system, which hands off the new position to the player 106. The player 106 then passes this new position to the core DLL 108, which passes it to the object. Now, the object knows not only its new position, but also the new relative positions of all the sub-objects within the object.

The ARTDOC document supports a number of different objects. In general, three types of objects exist: those with a visual representation, those with an audio representation, and those with specialized functions. Examples of objects with visual representation include images, which are generally static bitmaps. Objects with audio representation include digitally sampled sounds, which may be WAV or AIFF files, and MIDI files, which are usually played through an FM synthesizer. Specialized objects include frame objects, separator objects, and hyperlink objects. Frame objects hold multimedia objects, such as images, sampled sounds, and MIDI files, and also may provide a visual representation of the document. Separator objects provide a visual horizontal line (of different styles) that either separate paragraphs of text (moving with the text as it is reformed), or may exist at a geometrical position in the document. Hyperlink objects allow navigation to other content described by the author of the document.

also supports ART file objects, which are described in copending U.S. Patent Application Serial No. 08/755,586, assigned to the assignee of the present application. ART file objects include slide shows, sound and picture objects, ART sounds, and ART images. Any type of document can be contained inside an ARTDOC document by creating an ARTDOC extension module for the object.

Each object may include some kind of command for perception. Typically the perception of the object will be displaying of video or listening to sound. The command for perception will be a command indicating the kind of player that is used to produce the perception. If the player is not resident in the playing computer, various steps are taken as described herein.

In addition, each object may be able to pass and receive messages and to supply and retrieve its data. As an example of these functions, suppose the author wants to create an ARTDOC document that includes a video object (which includes image and audio portions) from a third party library. The library for the video object is "wrapped" with an ARTDOC document module, effectively creating an ARTDOC extension. This "wrapper" module allows the ARTDOC system to communicate with the third party library so that it can become part of the ARTDOC document. The wrapper module provides the translation layer and interprets and provides messaging between the ARTDOC document and the third party library. Referring to FIGURE 1, as the data stream 102 is received, the data stream meant for the video object is extracted and passed to the ARTDOC video object that wraps the third party library. The wrapper module then passes the data to the library, which interprets the data. The ARTDOC document cannot interpret the data, because it only carries the data. The library then interprets the data and provides, for example, a new video image and sound bite back to the wrapper module. The wrapper module, in turn, communicates back to the ARTDOC document, which displays the image at the proper place in the document. The video object may communicate with sound hardware directly to provide the sound. The system may work in this manner for all objects, including the text that is passed to the rich text processor and the data that is passed to the player 106.

When authoring an ARTDOC document, the user may produce some text using the Paige® system. Paige® will place the text on the computer display according to the desired type of rich text characteristics. Now, if the user wants to place an image within the ARTDOC document, the user employs a mouse, for example, to drag and drop the image into the ARTDOC document. The computer operating system that controls and monitors mouse operations passes the new information indicative of the mouse information (*e.g.*, the drag and drop information) into the document 202 shown in FIGURE 2.

The document 202 forms a code segment that receives image information according to the flow diagram of FIGURE 4, which constructs the particular frame 206. The process 400 begins at step 402, where an image, *e.g.*, a GIF image, is dropped into an ARTDOC document. The operating system sends information about the image to the player wrapper 106, which notifies the ARTDOC document at step 404. The information itself is then passed to a decoder at step 406. The decoder determines the data format (here GIF) and provides size information and visual representation back to the ARTDOC document at step 408. The ARTDOC document encloses the image in a frame and adopts the component object as a new child in step 410. The position of the ARTDOC document may be given by the original operating system mouse location.

An important feature is that the objects forming the images may be generic. Thus, all objects have the same characteristics and can be connected together to form parts of the ARTDOC document. The preferred items forming the objects include a framed image, a slide show, framed text, sound, a separator, and a hyperlink. Each of these will now be described in detail. Many of these objects may be provided by the player 106 as an ARTDOC document extension.

Frame objects (e.g., frame 206) are containers that perform several tasks in an ARTDOC document. First, they define the exclusionary areas within the page around which the text will wrap. Second, they contain the multimedia types described above, *i.e.*, images, sounds, video, *etc.*, or text objects, thereby creating framed text objects. Each of these multimedia types and text objects may also have further children, as described above. Finally, frame objects may provide a visual representation as an enclosing border at the author's option. Examples of visual representation options are color, thickness, pen style, outer margins, and three-dimensional effects, which can be fully implemented by the ARTDOC document.

Image objects (e.g., image 214) have an original size and provide a static visual representation. The author may override the original size. Image objects may be contained in frames or may appear in-line in a block of text and may be provided by the ART player 106 as an ARTDOC document extension.

Sound objects do not have a size, but are given the appearance of a sound icon if they are authored without a visual component. Sound objects may be contained in frames (e.g., sound file 219) or appear in-line in a block of text (e.g., in-line sound file 230). Sound objects may be of the sample wave type or of the sequenced MIDI type. Sound objects may be played by clicking on the visual representation or by other direction of the user and may also be provided by the ART player 106 as an ARTDOC document extension.

Sound and picture objects are a combination of a sound component and an image component, giving the visual representation of an image, but allowing the sound to be played when clicked. Like sound and image objects, sound and picture objects may be provided by the ART player 106 as an ARTDOC document extension.

Non-static multimedia objects include slideshows (e.g., slideshow 217) and video, as well as any similar moving-image-with-sound type of representation. Non-static multimedia objects are displayed as an image (optionally with a title page, if supported) until the user clicks or directs the object to be played, which results in a changing image, usually accompanied by sound. These objects may also be provided by the ART player 106 as an ARTDOC document extension.

Textual objects may appear in a frame (sometimes referred to as framed text objects, e.g., text 222); they may also appear in the form of the main body text (e.g., main text 204) and as captions for images (e.g., caption text 218). Text objects flow within the boundaries of the enclosing object and are stored in the file as an RTF data stream with embedded hyperlinks. Textual objects are implemented as an ARTDOC wrapper around the Paige® text processor.

Separator objects (e.g., hypertext 208) are used to provide a visual separation between paragraphs of text or other objects. Separator objects allow all the visual representation options as do frames, but do not contain objects. Separators are often used with "Anchor" geometry, which will be described below. Separator objects may be fully implemented by the ARTDOC document itself.

Finally, hyperlink objects (e.g., hypertext 208) are associated with any of the multimedia object types (images, sounds, slideshows, *etc.*) or with textual objects. When associated with multimedia objects, the hyperlink is described as a geometric (preferably rectangular) region on the associated object. When associated with text, the text appears visually different according to the author's specification. The user receives feedback when moving the mouse over a hyperlink region or hyperlink text, which includes a description of the location to which the hyperlink points. When the user clicks on the region or text, the appropriate information is retrieved and displayed for the user, or the user is taken to the location designated by the hyperlink pointer. Hyperlink objects are also fully implemented within the ARTDOC document.

The architecture of the ARTDOC document is fully extensible, meaning that the document will support any kind of multimedia object. This includes image, sound, video, and text stream object. With any multimedia object, the data for that object will be delivered and progressively displayed or played.

The "geometry" of an object refers to the specification for the dynamic positioning of the object within the bound of its parent or controlling object. Any visual object in an ARTDOC document has four edges: left, top, right, and bottom. The boundaries of such an object can be fully determined by applying a single "rod," "spring," "measure," or "anchor" to each edge. A rod is a fixed delta in parent coordinates from an edge of a child object to the same edge of its parent object. A spring is a percentage of the parent's total width or height from which the position of the child's edge should be determined. Springs calculate from the center of the object when paired with a measure; otherwise, the position of each edge is calculated independently. An anchor is a fixed delta in parent coordinates from the bottom edge of the first character of a paragraph to the top edge of a child object. A measure is the length of an edge (width or height) of an object and is used to fix the width or height.

The most common geometry is that of a fixed-position object that is described using a rod for the top and left sides of the object and a measure from the bottom and right sides. An object with this geometry retains its relative position form the top, left corner of the controlling object (in this example, the parent document) and also does not change size when the document is resized. An object with horizontal centering geometry is created by using a spring (set at 50%) for the left edge and a measure for the right edge. This causes the position of the object to move as the document resizes (remaining in the center), but still does not change the size of the object.

Other applications may include a geometry that attaches an object to the right edge of a document by using a left-edge measure and a right-edge rod. Allowing an object to grow with the document is also supported by attaching rods to both sides and not using measures, thus removing the fixed-measure constraint.

The anchor geometry setting allows an object's vertical position to be dependent on the position of a block of text. This setting is most often used with separators to ensure that they remain between the blocks of text that they separate, but can be used with any object.

III. The ARTDOC Document File Format

FIGURE 5 shows the preferred ARTDOC document file format 500 which includes a header area 502, an object archive 504, and a multiplex section 516. The header area 502 includes typical header information, such as start bits and version number, and any other overhead information.

The object archive 504 follows the header area 502 in the file format 500. The object archive 504 stores object information, but does not store the data associated with that information. The object archive 504 includes the information from the hierarchy of FIGURE 2. Each element in the hierarchy 200, from top to bottom, is used to provide an explanation of the information about each element of the hierarchy and its children. In the preferred embodiment of the present invention, the document is traversed in a "depth-first" manner, meaning that the layout and attributes of each element of the document are provided in the object archive 504. Consequently, the object archive will include information about the geometry (layout, position, and size, as described above) of the document, as well as complete descriptions of the document attributes, such as the frame.

The object archive 504 is the first information received by a receiving element. Hence, the receiver can display layout information about an ARTDOC document as soon as the receiver has received the object archive 504. As explained above, however, the data to fill that layout is not received with the object archive 504.

A embodiment has several features

First, this embodiment allows a first portion of an object's data to be a splash or "miniature representative rendering" of the object. Whether an image has a splash is determined by the particular format of the object in question. The splash data, if present, appears at the front of the image stream in order to provide the splash early in the download. The splash image indicates some very coarse information about the content of the multimedia objects in the ARTDOC document. The splash information is progressively updated as more data arrives that describes the images. More items of image information can be provided to the object to display more information. Text can also arrive and be displayed.

The data for each of the multimedia objects (*e.g.*, images, sounds, text streams, or video) is delivered in the multiplex section 516 of the ARTDOC file 500. The data is generally the same data that would be resident in a file containing the object alone, without any of the geometry and attribute information, as this information is contained in the object archive 504. The multiplex section 516 represents the bulk of the data stream.

In the multiplex section 516, data is interleaved together according to the "choreography" of the document. The term choreography, as used herein, refers to the ordering of the multimedia objects in the ARTDOC file format. When objects, such as images, are designated to appear during the same time interval, their data is interleaved together in the multiplex section 516. As the ARTDOC file 500 is received, the interleaved data stream is decomposed, and the appropriate data is delivered to each object, progressively updating each object's visual appearance in the display of the ARTDOC document.

Some classes of objects, however, are not interleaved with other objects in the multiplex section 516, regardless of the author's choreography designation. The first class includes those objects that cannot be progressively rendered. MIDI files and most standard types of audio and video files are examples of this class of objects. Because these objects must be completely downloaded before they can be interpreted, no benefit flows from interleaving their data. The second class of non-interleaved objects includes files that are designed to be played back progressively, but that are authored for a particular bandwidth. These objects are referred to as time-based or "temporal" files. Examples of such temporal files are certain audio and slide show files that cannot be interleaved with other data without slowing the delivery of their own temporal data and risking starving their players of data.

The classes of objects that are not interleaved are detected by the ARTDOC document and are placed in their own choreography groups within the multiplex section 516. These groups may be arranged with interleaved multiplex groups without degrading document playback, as interleaved and non-interleaved groups do not overlap.

By interleaving certain data, such as images, and separating temporal files, the present invention permits the incremental "display" of the document (including sounds and images) to the user. As described above, images can be multiplexed or mixed together so that they arrive and are displayed at substantially the same time. The constraints of narrow-band, *i.e.*, modem communication, however, generally prevent interleaving of the temporal objects, the quality of their playback would likely suffer if they were arbitrarily mixed together. None of the proposed encapsulated formats for HTML allows such an incremental display and do not allow for the proper organizing of temporal information for narrow-band applications. The proposed HTML formats rely on broad band communication, meaning that these formats may arbitrarily mix the data, regardless of the type of data, and may sequentially store objects, sacrificing either the audio or video presentation of the HTML document.

It is known that particular object data, such as text streams, may not be compressed. The ARTDOC document can compress such data itself into the multiplex section 516. If this is done, the compression flag (see reference numeral 708 of FIGURE 7) is set.

An unknown object is a special type of object within the ARTDOC format. The unknown object is an object that probably has a defined player, but the computer receiving the unknown object does not recognize the defined player. The unknown object includes the information contained in the object archive 504 (described above), because the unknown object is a type of object. Therefore, the receiving computer knows where this information is and how large it is. Hence, the player 106 can draw the outlines of the unknown object, without putting the particular details in it. At least some of the data from the multiplex section 516 is provided to the unknown object, which displays an "unknown" icon. Although that data cannot be displayed, it can be carried around by the file embodying the unknown object, and the file can be saved intact. Moreover, later downloads of coded information can be made to allow playing of the unknown object. For example, the unknown object may reference a DLL of which the playing system 106 has no knowledge. Later download of the DLL may allow later display of the unknown object.

The ability to support unknown object types also allows the use of user-defined object types. So long as the object type follows the conventions given above, it can be made into part of the ARTDOC format, and encapsulated within the ARTDOC document.

IV. The Multiplexed Scheme

The preferred embodiment uses a multiplexed scheme, as noted with respect to FIGURE 5. The multiplexed scheme preferably employs "slices" of information. Each slice represents a piece of information that can make some degree of difference in the perceived ARTDOC document. A slice may be a segment of audio between any two natural pauses in the audio, a piece of an image that causes a further update of the image, or a block of text.

FIGURE 6 shows the preferred multiplex message 516 in accordance with the multiplexed scheme The multiplex message 516 includes an object number counter 602. The object number counter 602 includes an indication of the number of "objects" that will follow in the object section of the multiplexed information.

Object descriptions 604 follow the object counter 602. The object descriptions 604, being of the same number as the object number counter 602, include an object reference 606 that provides a back reference to the object information stored in the object archive 504 and a flag 608 indicating whether the object data was compressed by the ARTDOC document.

Following the object descriptions 604 are choreography groups 610. There may be any number of choreography groups 610 (as described below) corresponding to the ordering of interleaved data within the document. When no further groups exist in the multiplex section 516, a single zero-byte 624 is used to mark the end of the multiplex section 516.

A choreography group 610 includes an object counter 612 that indicates the number of objects contained in the particular choreography group 610. Objects may appear in different groups, providing a portion of their data from within each group. Following the object counter 612 is a section 614 providing the size and type information for each object in the choreography group 610. The size and type section 614 provides an internal characterization of the multiplex data as well as the total length of the multiplex data for the object contained in this particular choreography group 610. An object header information section 616 follows the size and type information section 614. The object header section 616 includes a section 618 that contains the length of the object's data slice within the group 610 and an object reference section 620 that provides a back reference to the object's description 604. The data slice length section 618 is an optional parameter, because, if only one object is contained in the group 610 (*i.e.*, the object counter 612 indicates "1"), the slice length is assumed to be the same as the length information provided in size and type section 614.

The actual interleaved data 622 follows the object header information 616. The interleaved data 622 constitutes the bulk of the information included in the choreography group 610 and includes the data destined for the referenced object. The interleaved slice data 622 is preferably provided as a series of raw data bytes. The length of the interleaved slice data is given by the data slice length section 618 for each object in the choreography group 610. Each object supplies its indicated length of data in turn, repeating until completion of the full amount of data given by the size and type section 614 for each object. The last slice of a particular object may be shorter than the size given by the size and type section 614. In such a case, the last slice will correspond to the remaining total bytes to be delivered in the group 610, as indicated by the size and type section 614. Reference numeral 626 is an examplary interleaved slice data section for two objects. If the object count 612 indicates "1", there is only one object in the group 610 and only one slice of data of the length given by the single entry in the size and type section 614. These single object groups represent non-interleaved data, as described above.

The author may place objects or portions of objects into the choreography groups 610 by means of the user-interface. By placing an object into the same choreography group as another object, the author directs that those objects will be delivered interleaved together. Consequently, those interleaved objects will appear progressively over the same duration during the file download. Objects placed in later choreography groups will appear in time after objects placed in earlier groups. The data for a particular object may be placed in different choreography groups to create, for example, the effect of having only the initial splashes of images appear early in the document, followed by the text of the document in a later group, and finally by the remainder of the image data in even later groups.

An example of choreography for an ARTDOC document 700 is shown in FIGURE 7, which illustrates three displays of the document 700 over time. In this embodiment, portions of the entirety of the ARTDOC document 700 can be sent simultaneously. A first part of the ARTDOC document 700 is displayed at time T_1 as a "splash" of all images 702 and 703. This is followed by a second part of the document 700 at time T_2, including more information about the images 702, 703 as well as text part 1 704. Time T_3 shows the final ARTDOC document 700, with completed images 702, 703, as well as additional text parts 2 and 3, 706 and 708, respectively.

If the user does not specify a particular choreography, a default organization will be provided. The default organization preferably groups objects from top to bottom as they appear in the ARTDOC document. Moreover, in the default, objects that are located roughly near one another may be placed in the same group to ensure that they will be delivered together.

FIGURE 8 illustrates the default method for organizing an ARTDOC document 800. The ARTDOC document 800 is conceptually arranged into a number of areas, called "buckets." Each bucket can be a physical area on the ARTDOC document 800, or alternatively could be a portion of different areas on the ARTDOC document 800. FIGURE 8 shows the ARTDOC document 800 being broken up into four physical sections—Section 1 810, Section 2 820, Section 3 830, and Section 4 840. Section 1 810 includes both text 812 and an image I1 814. Section 2 820 includes text 822 and an image I2 824, as well as a portion 826 of image I3. Section 3 830 includes another portion 832 of image I3 and some text 834. Section 4 840 includes only text 842.

Each of these sections 810, 820, 830, 840 is treated as a bucket, and each bucket represents an area on the ARTDOC document 800. Each bucket is analyzed to determine slice sizes that make sense. An appropriate slice should be big enough to alter the perception presented to the user, but small enough to enable simultaneous occurrence of events.

Bucket 1 (*i.e.*, Section 1 810) has two objects including the text portion 812 and image I1 814. Bucket 2 (*i.e.*, Section 2 820) has three objects, the text 822, the image I2 824, and the portion 826 of image I3. Bucket 3 (*i.e.*, Section 3 830) has two objects, the other portion 832 of image I3 and the text 834, and Bucket 4 (*i.e.*, Section 4 840) has only one object, the text 842. Therefore, the system includes multiple areas forming the ARTDOC document 800.

The integrated file format of the present invention allows all parts of the image to be received in sequence. The sequence can be set by the author of the document being perceived. The sequence is also such that at least part of the sequence allows portions of the information in the document to be progressively received.

ABSTRACT

Methods for producing and multiplexing a file format, as well as structures for a hierarchical file format and data file, are provided. The data file includes data that is divided in a hierarchical manner, including a highest level document portion that supports all lower level portions of the data file. The hierarchical data file forms a multimedia document that can be displayed on a computer display with accompanying audio. The multimedia document may include data in a variety of file formats, including image data, sound data, textual data, and video data. At least some of the data is multiplexed in the data file, so that the multiplexed data can be progressively played and displayed as it is downloaded by a computer. Data that cannot be progressively played need not be multiplexed in the data file and can be located in an area of the data file separate from the multiplexed data.

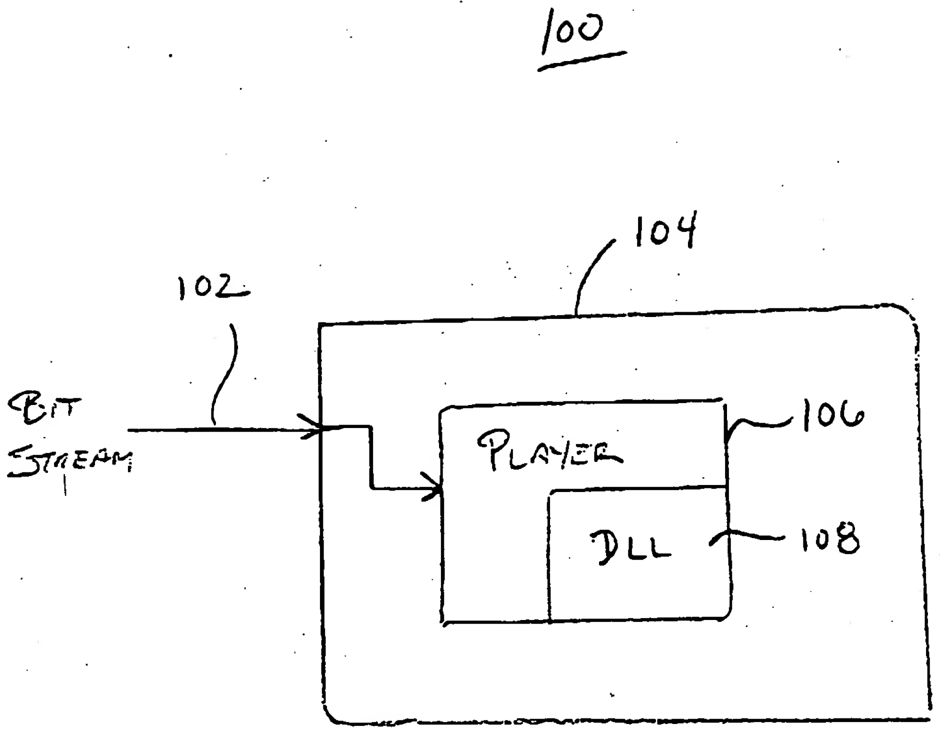
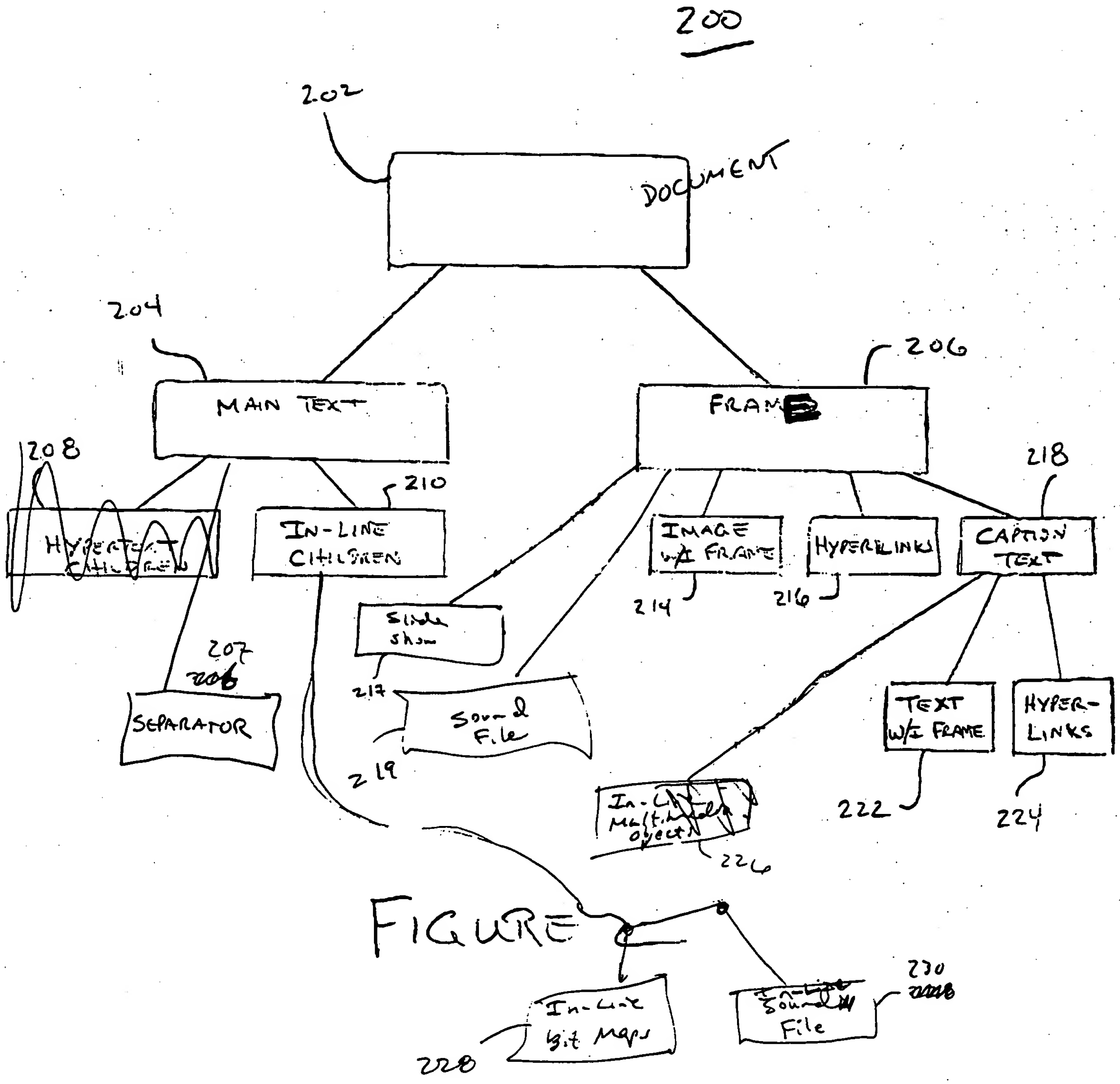
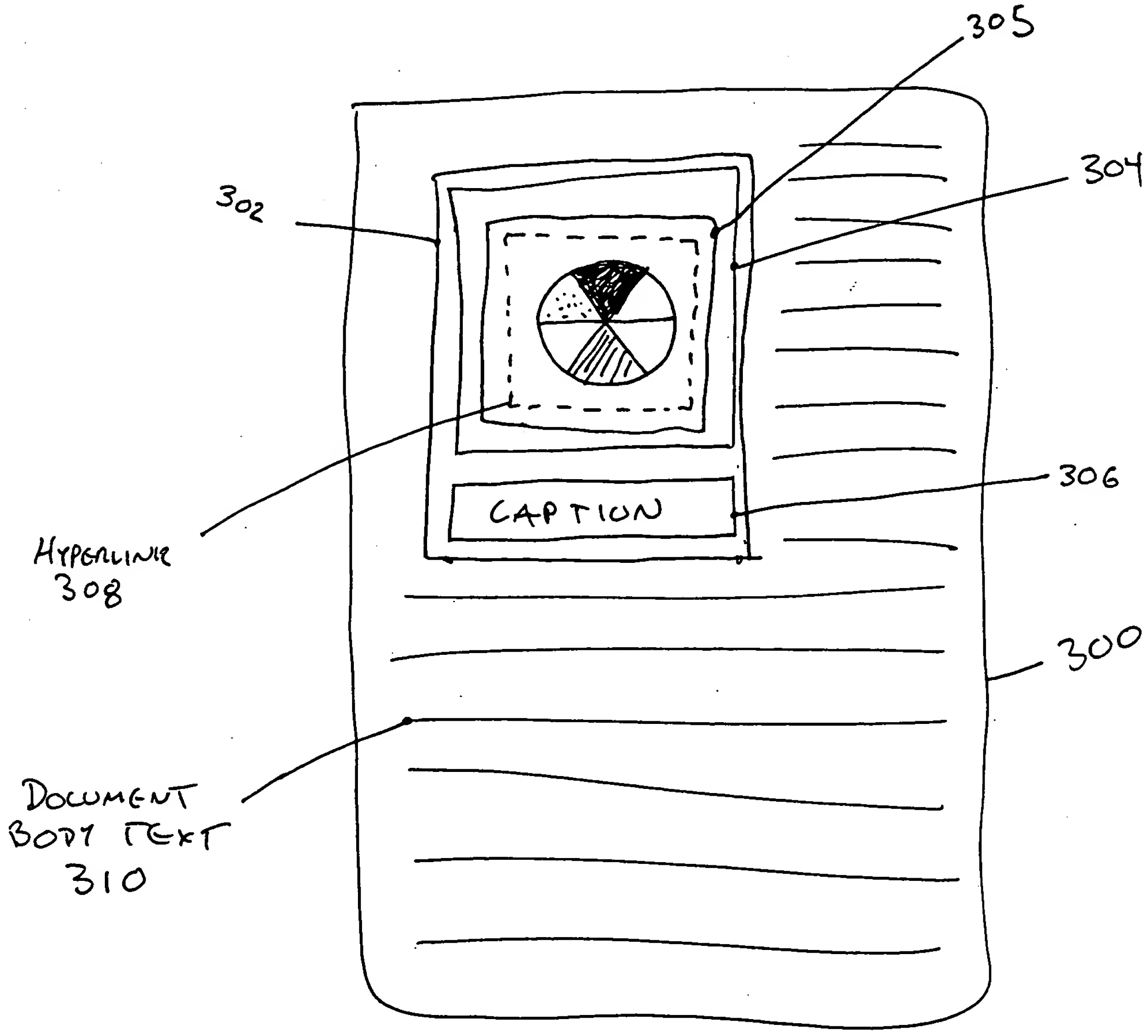


FIGURE 1


200
202
DOCUMENT
204
MAIN TEXT
206
FRAME
208
HYPERTEXT CHILDREN
210
IN-LINE CHILDREN
207
~~206~~
SEPARATOR
Slide show
217
Sound File
219
IMAGE w/i FRAME
214
HYPERLINKS
216
218
CAPTION TEXT
TEXT w/i FRAME
222
HYPER-LINKS
224
In-Line Multimedia object
226
FIGURE
In-Line bit Maps
228
~~In-Line~~ Sound File
230



305
302
304
CAPTION
306
HYPERLINK
308
300
DOCUMENT
BODY TEXT
310

FIG 3

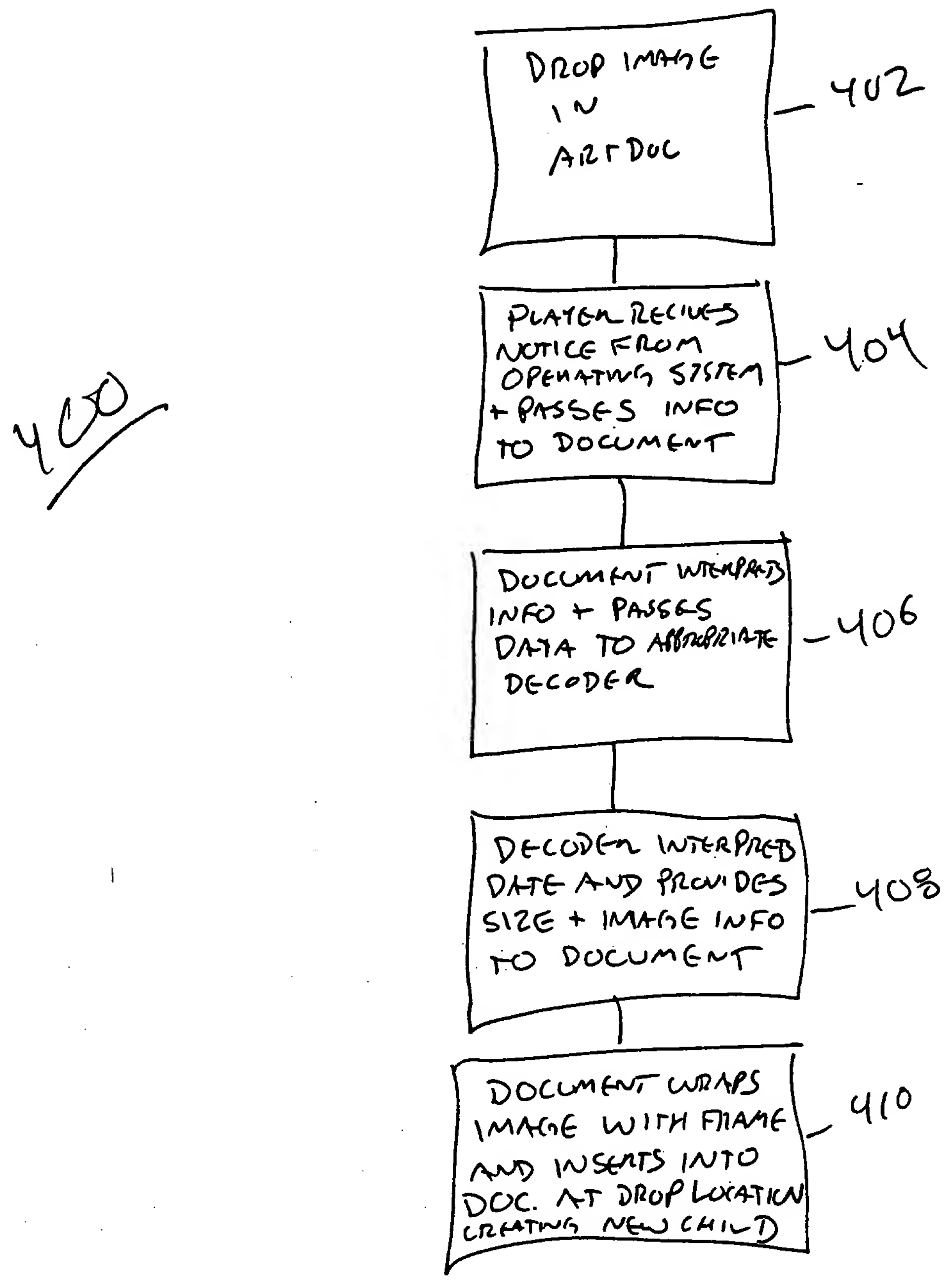

DROP IMAGE IN ARTDOC
402
PLAYER RECIVES NOTICE FROM OPERATING SYSTEM + PASSES INFO TO DOCUMENT
404
DOCUMENT INTERPRETS INFO + PASSES DATA TO APPROPRIATE DECODER
406
DECODER INTERPRETS DATE AND PROVIDES SIZE + IMAGE INFO TO DOCUMENT
408
DOCUMENT WRAPS IMAGE WITH FRAME AND INSERTS INTO DOC. AT DROP LOCATION CREATING NEW CHILD
410
400

FIG. 4

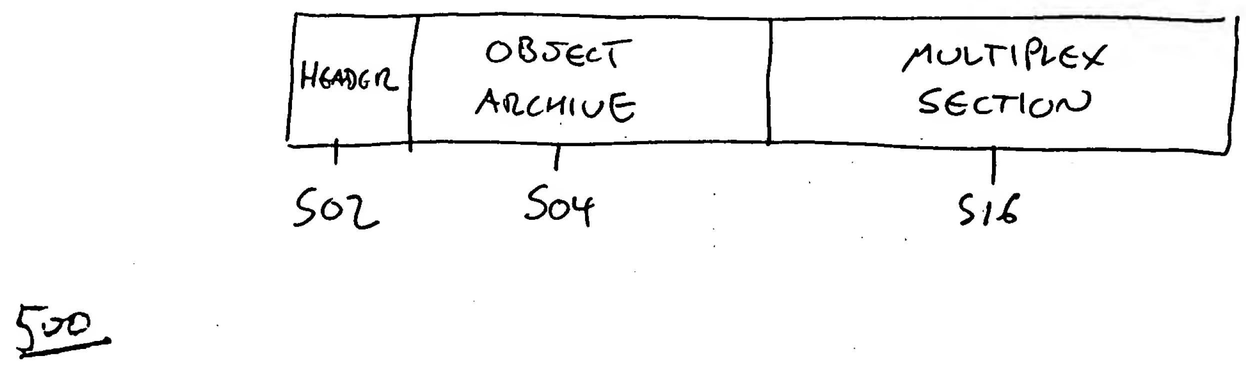
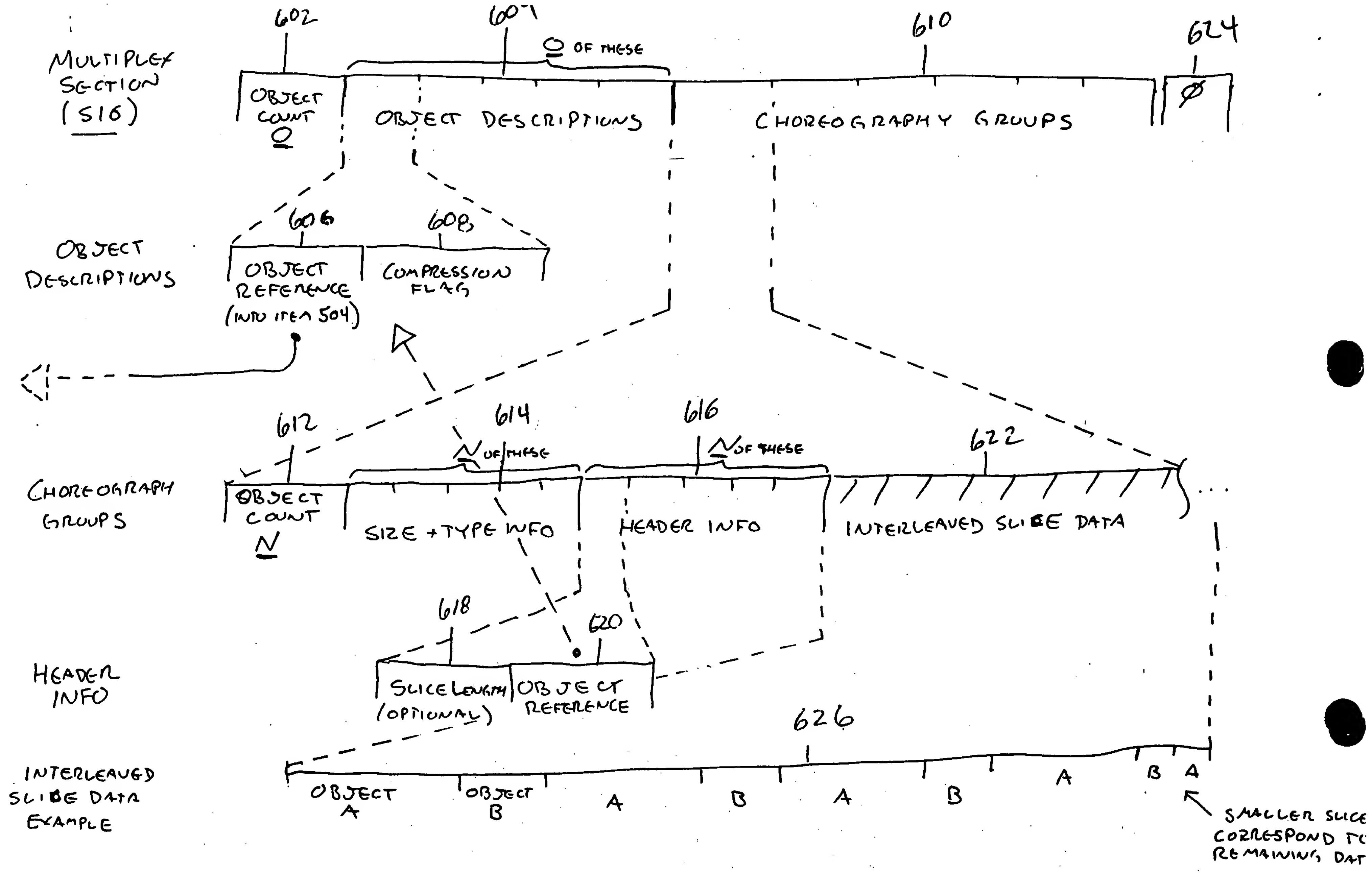


FIGURE 5



MULTIPLEX SECTION (516)
602
604
O OF THESE
610
624
OBJECT COUNT O
OBJECT DESCRIPTIONS
CHOREOGRAPHY GROUPS
Ø
OBJECT DESCRIPTIONS
606
608
OBJECT REFERENCE (INTO ITEM 504)
COMPRESSION FLAG
CHOREOGRAPH GROUPS
612
614
N OF THESE
616
N OF THESE
622
OBJECT COUNT N
SIZE + TYPE INFO
HEADER INFO
INTERLEAVED SLICE DATA
...
HEADER INFO
618
620
SLICE LENGTH (OPTIONAL)
OBJECT REFERENCE
INTERLEAVED SLICE DATA EXAMPLE
626
OBJECT A
OBJECT B
A
B
A
B
A
B
A
SMALLER SLICE
CORRESPOND TC
REMAINING DAT

FIGURE 6

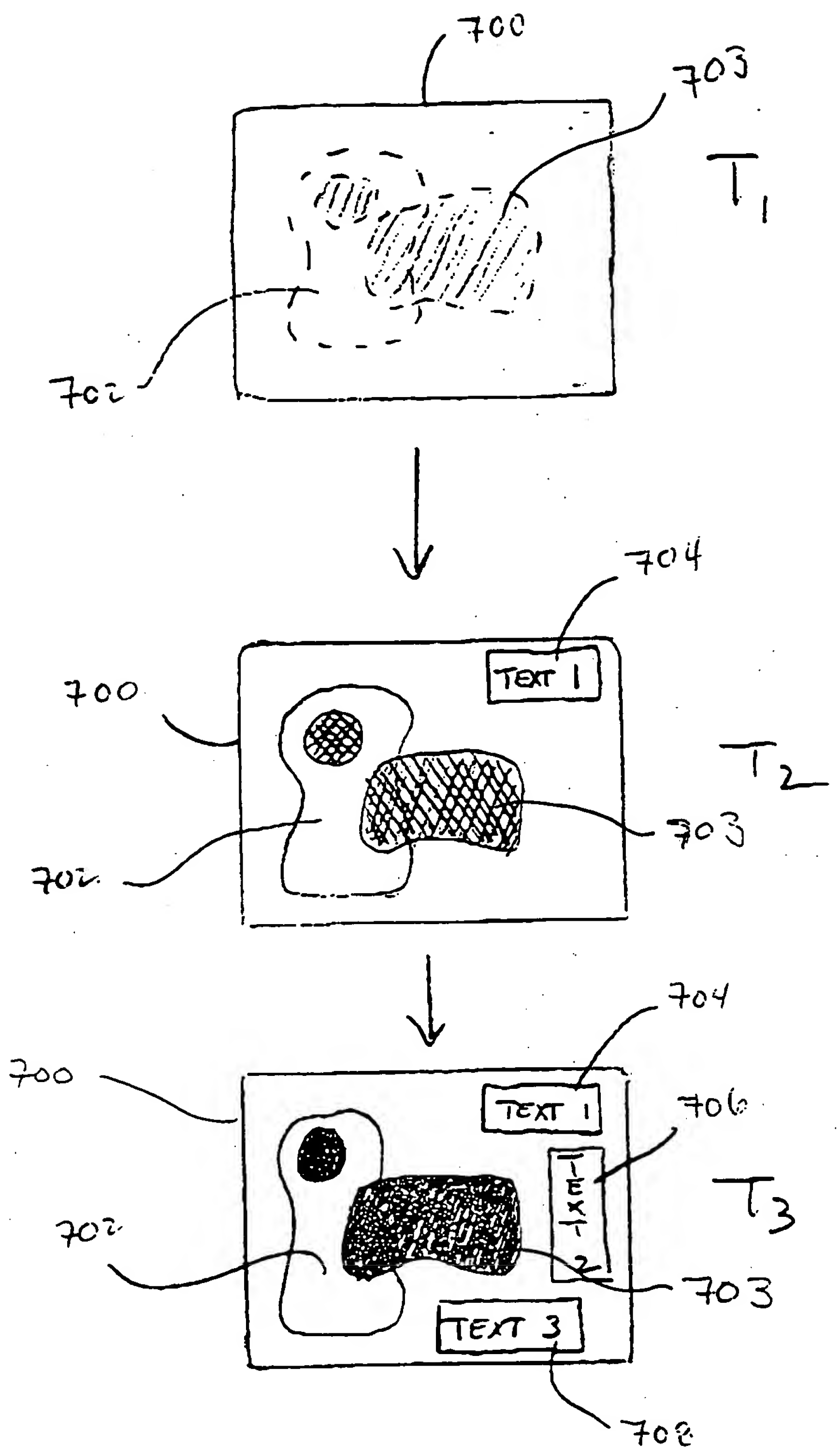

700
703
T1
702
704
TEXT 1
700
T2
703
702
704
TEXT 1
700
706
TEXT 2
T3
702
703
TEXT 3
708

FIGURE 7

```
*********************
***   TX REPORT   ***
*********************

TRANSMISSION OK

TX/RX NO                 3992
CONNECTION TEL           00800001#17148234444
SUBADDRESS
CONNECTION ID
ST. TIME
USAGE T                  14'39
PGS.                      44
RESULT                   OK
```

FISH & RICHARDSON P.C.

4225 Executive Square
Suite 1400
La Jolla, California
92037

Telephone
619 678-5070

Facsimile
619 678-5099

Date	
To	David Corboy America Online, Inc.
Facsimile number	
From	
Re	
Number of pages including this page	43

FISH & RICHARDSON P.C.

4225 Executive Square
Suite 1400
La Jolla, California
92037

Telephone
619 678-5070

Facsimile
619 678-5099

Date	
To	David Corboy America Online, Inc.
Facsimile number	
From	
Re	
Number of pages including this page	43